Using the GCC toolchain for Mulle SW development.

Tested on Windows XP and Mac OS X Snow Leopard

© 2011 Eistec AB

All rights reserved. Subject to change without prior notice.

Document version 4.00

1 ENVIRONMENT SETUP

1.1 NECESSARY FILES

First, download the following components:

- For Windows: MinGW and the MSYS environment from MinGW (including make), available at http://www.mingw.org/. Choose the GUI installer[1].
- Eistec's precompiled GCC compiler for M16C, consisting of two compressed archives. These archives are available at Eistec's home page, http://www.eistec.se/software.php.
  - For Windows XP, download:
    m32c-elf-binutils-2.19-i586-mingw32msvc.tar.bz2
    m32c-elf-gcc-4.4.1-i586-mingw32msvc.tar.bz2
    Furthermore, a rar utility is necessary in order to decompress the archives. WinRAR is recommended, while WinZIP is known NOT to work with these files.
  - For Mac OS X, download:
    m32c-elf-gcc-4.4.3-OSX-10.6.2-i586.tar.bz2

1.2 INSTALLATION AND SETUP UNDER WINDOWS XP

Begin by launching the MinGW installer. Choose "Download latest repository catalogues". Install in "C:\MinGW" as set by default. Be sure to include "MSYS Basic System" as shown in Figure 1 below.

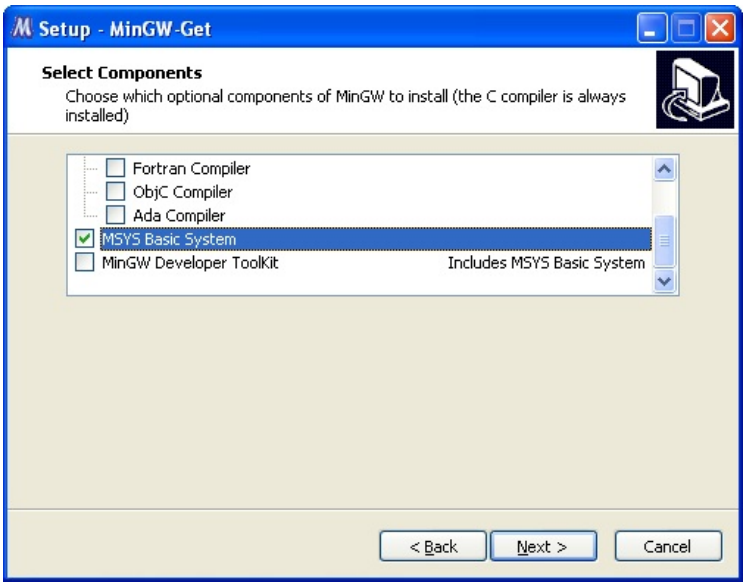

Figure 1: MinGW installation

[1] http://sourceforge.net/projects/mingw/files/Automated%20MinGW%20Installer/mingw-get-inst/mingw-get-inst-20110211/mingw-get-inst-20110211.exe/download

Next, extract the contents of the archives m32c-elf-binutils- and -gcc to a suitable folder. (In this example they are extracted to "`C:\Dev\Util`".) A folder named "`i586-mingw32msvc`" will be created in the chosen directory. When asked if you want to replace the existing file "`libiberty.a`", choose yes.

After installation, a few system variables have to be added in Windows. Open "System Properties", the "Advanced" tab, and click on "Environment Variables" (shown in Figure 2).

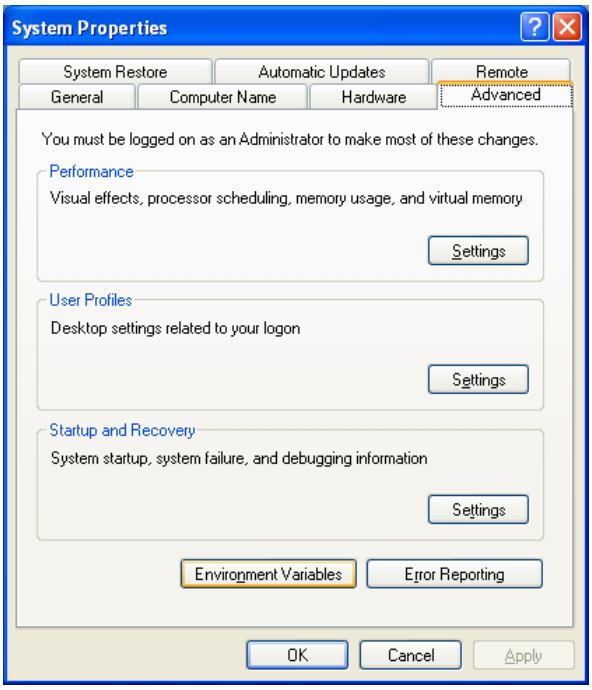

Figure 2: System properties

Find and select the System Variable "PATH" and click "Edit", see Figure 3. Add to the text already displayed in "Value" the location of the gcc compiler directory, in our case "`C:\Dev\Util\i586-mingw32msvc\bin`". Also add the path to MinGW, i.e: "`C:\MinGW\bin`".

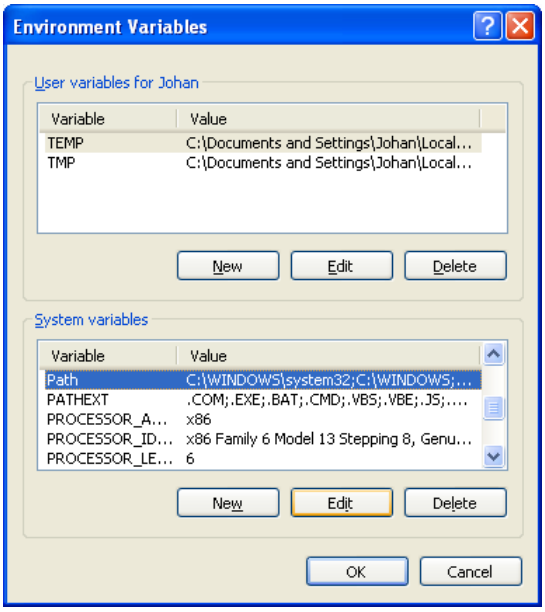

Figure 3: Environment Variables

1.3 INSTALLATION AND SETUP UNDER MAC OS X

First, install Xcode, available on your Mac OS X installation DVD or at developer.apple.com/mac/. Make sure to check "`UNIX Development`" as shown in Figure 4.

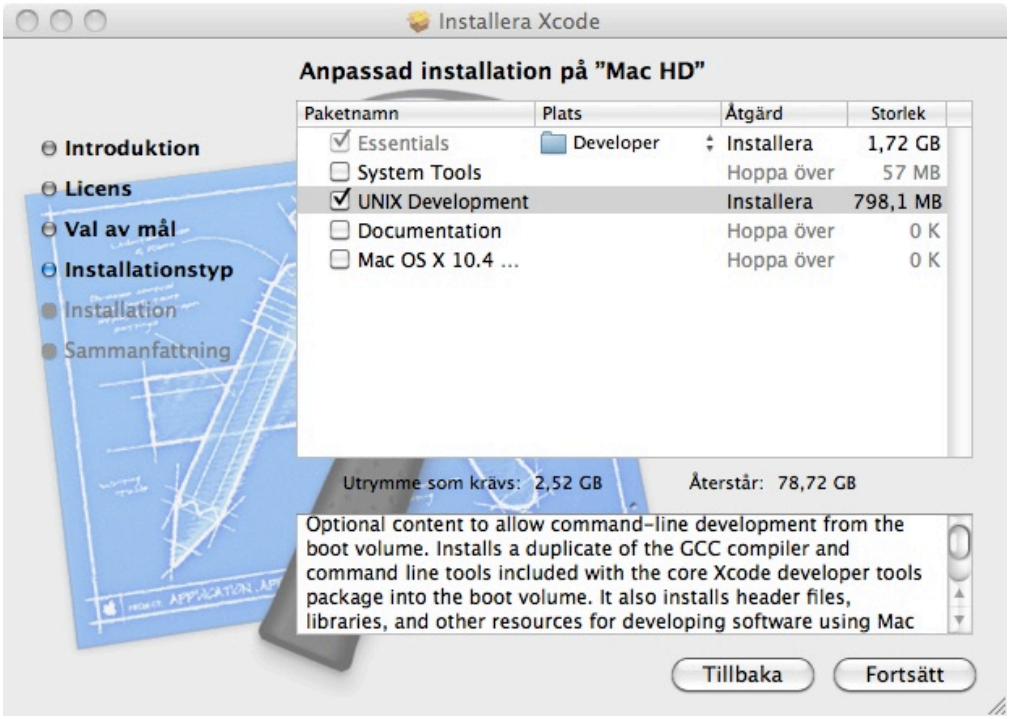

Figure 4: Xcode installation

Next, double click the gcc archive downloaded in chapter 1.1 to unpack it. Move to any directory of your choice. In this guide we put it in "`/Library`". The path to the compiler must be added to your `PATH` variable. Open a terminal window and type:

```
echo 'export PATH=/Library/opt/m32c-elf/bin:$PATH' >> ~/.profile
```

exactly as above, where "`/Library`" is where you have put the unpacked archive. Quit the Terminal application for the changes to take effect, before continuing (chapter 1.4).

1.4 CHECK YOUR INSTALLATION:

Open the Terminal window (on Mac) or the MinGW shell from the Start menu (on Windows XP). A shortcut to this file can be created on the desktop.

Type `make`. The text in Figure 5 (Mac) or Figure 6 (Windows) should appear.

Type `m32c-elf-gcc -v`. The text in Figure 5 (Mac) or Figure 6 (Windows) should appear.

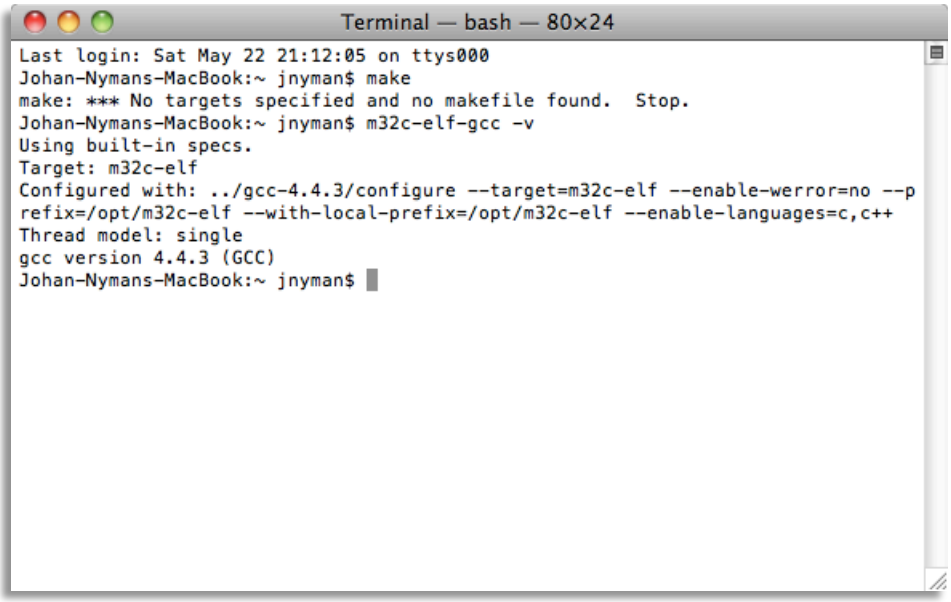
Terminal — bash — 80×24
Last login: Sat May 22 21:12:05 on ttys000
Johan-Nymans-MacBook:~ jnyman$ make
make: *** No targets specified and no makefile found. Stop.
Johan-Nymans-MacBook:~ jnyman$ m32c-elf-gcc -v
Using built-in specs.
Target: m32c-elf
Configured with: ../gcc-4.4.3/configure --target=m32c-elf --enable-werror=no --p
refix=/opt/m32c-elf --with-local-prefix=/opt/m32c-elf --enable-languages=c,c++
Thread model: single
gcc version 4.4.3 (GCC)
Johan-Nymans-MacBook:~ jnyman$

Figure 5: Installation test, Terminal (Mac OS X)

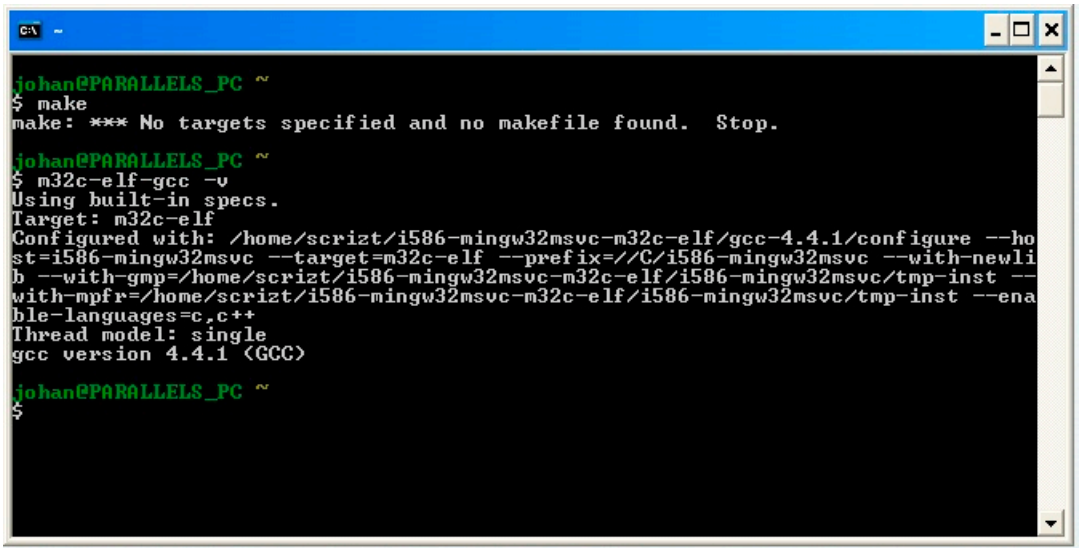

Figure 6: Installation test, MSYS (Windows XP).

If any other message appears, your installation may be incomplete. The system can probably not locate the correct executables. Check the environment variables by typing `$PATH`. If the path to the executables are not displayed, add them as described above.

2 USAGE

Edit the Mulle code in your favorite code editor or IDE.

Open MinGW Shell or a Terminal window and change to the directory of the current project.

Type “make clean” to remove old executables and object files.

Type “make all” to recompile the code and create a .mot file, downloadable to the Mulle.

See the document “Mulle Expansion Board User Manual” on how to download the compiled code to the Mulle Platform.

3 SET UP ECLIPSE (OPTIONAL)

Download and install the Eclipse IDE for C/C++ development[2].

Open the Eclipse IDE and create a new empty C project (“`Makefile Project/empty`” in the wizard). If any Makefile or c-file is created, delete them.

Right click on the project folder and choose “`Import…`”. In the wizard, select “`General-> File System`”, then next. Browse to the base directory of the Mulle SW project you want to import, and select it[3]. Tick the box in front of the base directory in order to import all files. The makefiles in the project then specifies the compiler and linker to be m32c-elf-gcc. Import all four parts of the Mulle Software, i.e. Applications, Library, lwBT and lwIP. Whether to place them in the same project or not is optional. In this guide they are four different projects.

Right click on the Applications project folder and choose `Properties` in the context menu. Click the `C/C++ Build` label. Make sure that the “`Generate Makefiles automatically`” and “`Use default build command`” checkbuttons are unchecked. Build command should be `make` when using Mac OS X, see Figure 7.

[2] http://www.eclipse.org/downloads/

[3] i.e. Applications, Library, lwBT, lwIP respectively.

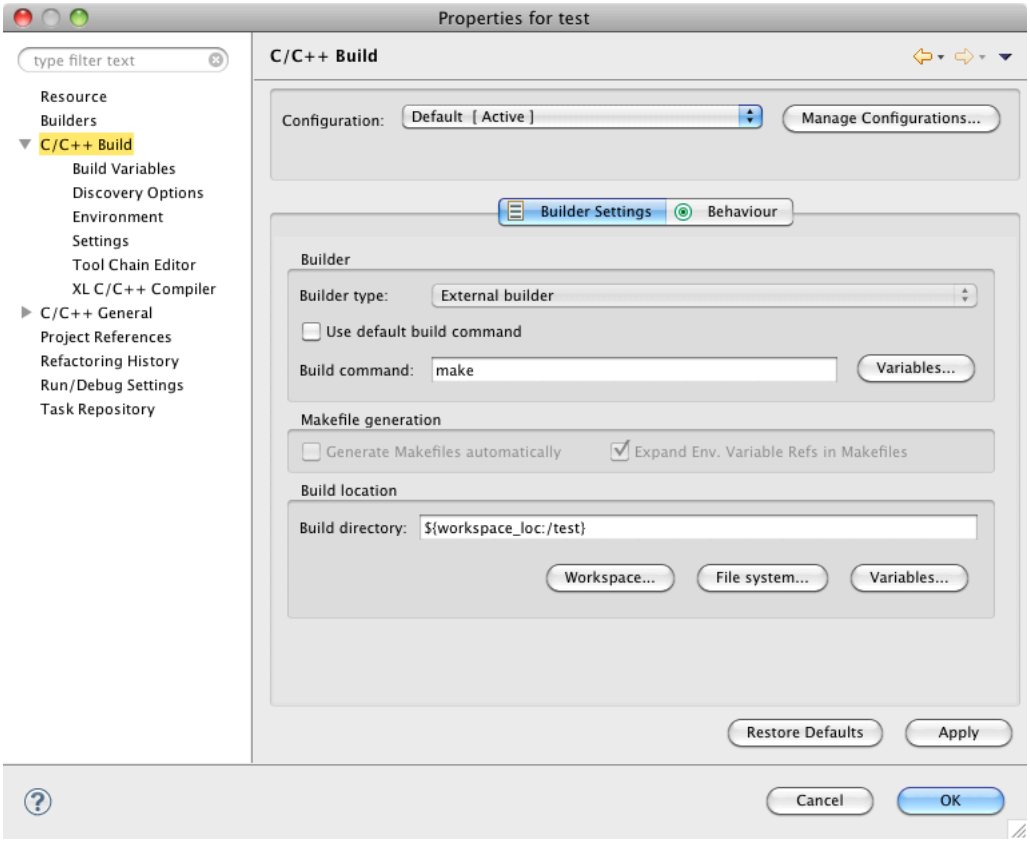

Figure 7: Eclipse Project properties (Mac OS X)

Under Windows XP, in order to work with your toolchain, the build command must be set to "`mingw32-make`", and configuration `GCC` selected as shown in Figure 8.

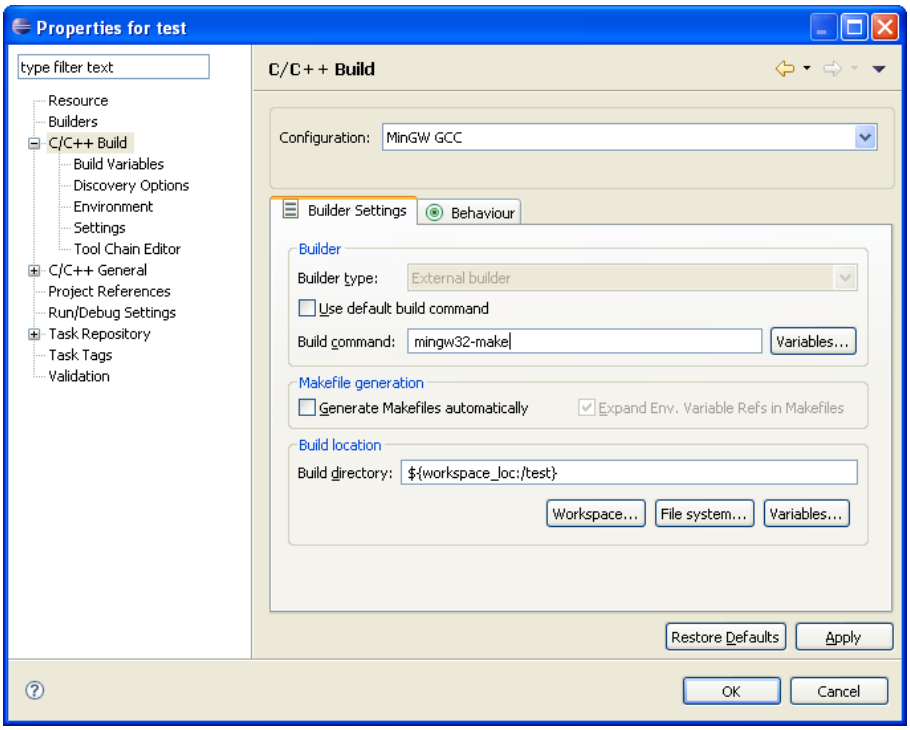

Figure 8: Eclipse Project Properties (Windows XP)

Go on and click "Environment", under the label "C/C++ Build". Choose to add a new variable. Name the variable "PATH" and enter the path to the m32c-elf-gcc compiler. Also make sure that the "Append variables to native environment" radiobutton is active before clicking OK. (See Figure 9)

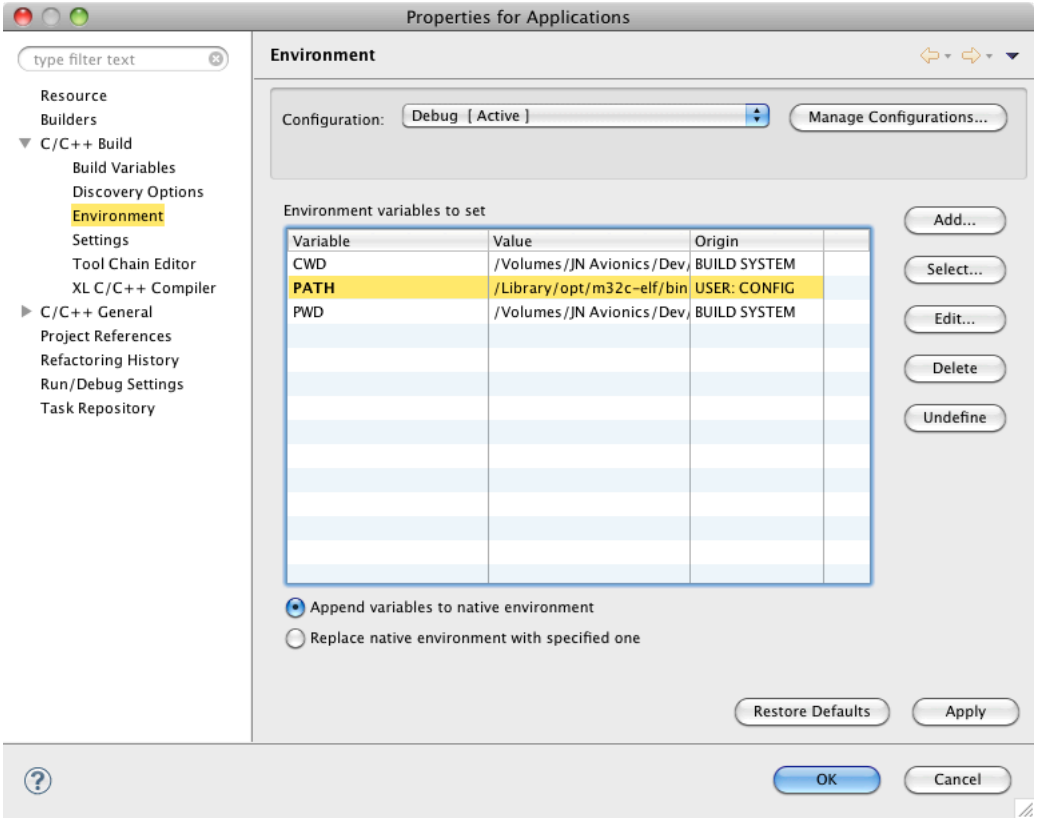

Figure 9: Adding the PATH variable to environment

IMPORTANT: Sometimes Eclipse rearranges the paths, and introduces a semicolon (;) which does not work as separator. If compiling does not work, i.e. an error message is produced saying "`make: m32c-elf-gcc: No such file or directory`", this is likely to have happened. In this case, edit the PATH variable as above, and make sure that the path to the m32c-elf/bin directory is stated first, followed by a colon (:), and thereafter your systems paths. No spaces should be used.

3.1 ADD MAKE TARGETS

On the right hand side of the eclipse window, in the "Make Targets" tab, go to the `Applications` source folder, right click on the project[4] and select "`New...`". In the dialog (Figure 10) choose a name for the target, e.g `All`. As make target type "`-f Makefile all`". Create another target, named "`Clean`", with the make target "`-f Makefile clean`".

[4] e.g. "Mulle_Demo"

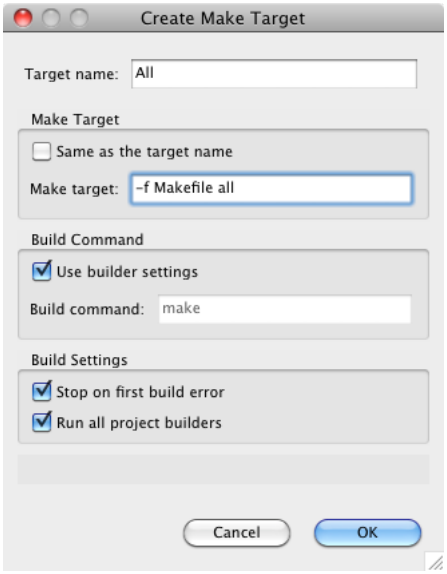

Figure 10: Eclipse create new make target dialogue

It could also be a good idea to uncheck the option "`Build Automatically`" in the `Project` menu of the Eclipse window.

Doubleclick the make target (the green circle) in order to build the selected project.